UISCEmod

UISCEMod User Manual

July 2023

Ollscoil Teicneolaíochta an Oirdheiscirt
South East Technological University

Geological Survey
Suirbhéireacht Gheolaíochta
Ireland | Éireann

South East Technological University

Geological Survey Ireland

UISCEMod User Manual

Joan Campanyà, Ted McCormack, Owen Naughton

Version 072023

Ireland, July 2023

Acknowledgments
UISCEmod was supported by Geological Survey Ireland [grant number: 2019-TRP-GW-001]. Data and expertise were provided by Geological Survey Ireland Groundwater and Geothermal Unit.

Disclaimer
Although every effort has been made to ensure the accuracy of the material contained in this manual, complete accuracy cannot be guaranteed. Neither South East Technological University, Geological Survey Ireland, nor the authors accept any responsibility whatsoever for loss or damage occasioned, or claimed to have been occasioned, in part or in full as a consequence of any person acting or refraining from acting, as a result of a matter contained in this manual or the use of UISCEmod.

Reference
UISCEmod User Manual

- Campanyà, J., McCormack, T., Naughton (2023) UISCEmod user manual.

UISCEmod code (use this reference if you use UISCEmod):

- Campanyà, J., McCormack, T., Gill, L.W., Johnston, P.M., Licciardi, A., Naughton, O., 2023. UISCEmod: Open-source software for modelling water level time series in ephemeral karstic wetlands. Environmental Modelling & Software 167, 105761. https://doi.org/10.1016/j.envsoft.2023.105761

CONTENTS

1. UISCEMOD SHORT DESCRIPTION

UISCEmod is an open-source software developed for modelling water level time series in ephemeral karstic wetlands. UISCEmod contains both experimental, EM, and lump conceptual, LM, hydrological models. The EM approach is based on a gamma distribution and treats the system as a whole, providing information about the behaviour of the hydrological system but without providing direct information about the individual components of the hydrological system. The LM approach is based on as storage model and distinguishes between inflow and outflow mechanisms, considers the dependency of the outflow with the depth of the flood, accounts for direct interaction with the atmosphere (direct rainfall and evaporation), and has the potential to incorporate specific information about the site (e.g. elevation of potential outflow mechanisms) into the model parameters.

The modelling of water level time series is, for both modelling approaches, based on input meteorological data, rainfall (R) and reference evapotranspiration (ET_o), which are combined to generate effective rainfall (ER) time series. ER time series are then considered to model water level time series. In the case of the LM approach, inflows and outflows from direct interaction with the atmosphere are also considered, including inflows from direct precipitation and outflows from evaporation (*Evap*). A summary of the main steps is presented in Figure 1. Modelling is performed with volume time series, instead of water level time series, to account for the stage-dependency of storage within the wetlands and to facilitate interpretation of the model parameters. Conversion from volume to stage time series, and vice-versa, is carried out using the stage-volume relationship for the specific site extracted from the Digital Terrain Models (DTM).

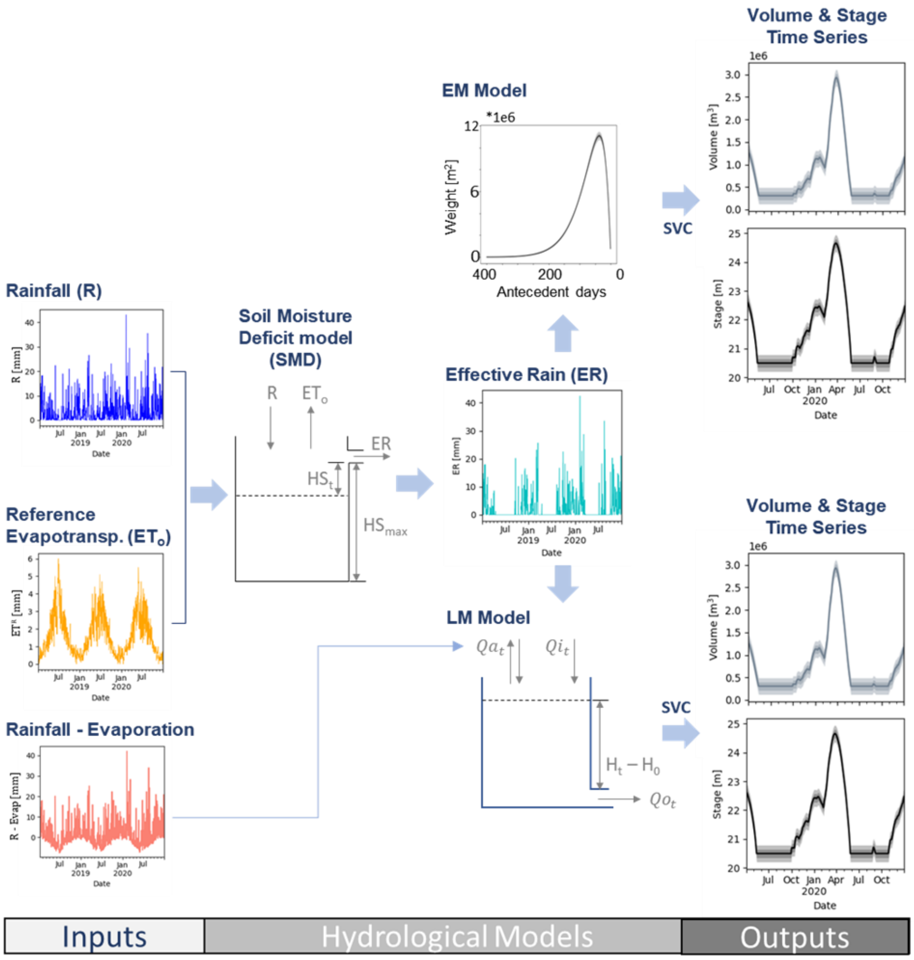

Figure 1 Conceptual Diagram of UISCEmod (from Campanyà et al. 2023[1]). R: Rainfall, ET_o: Evapotranspiration, SMD: Soil Moisture Deficit Model, HS:Elevation to outflow for SMD, ER: Effective rainfall, Q_i:Inflow ,Q_a: Inflow and outflow by direct interaction with athmosphere, H_o: Elevation of the outflow,Q_o:outflow., SVC: stage-volume conversion, EM: experimental model,LM: lump conceptual model.

In terms of the structure of UISCEmod, the code is divided by modules and functions to facilitate future updates and the use of the code in new areas and scenarios. A summary of the flowchart for UISCEmod is presented in Figure 2. The main modules are Input, Calibration, and Outputs. The Inputs module combines meteorological datasets as required by the hydrological models and generate variables needed for Calibration and Output modules. The Calibration module is used to calibrate the hydrological models and can be ignored if model parameters are already defined. The calibration process is automated following a Bayesian approach. The third module, Outputs, generates the main products including: forward solution for stage and volume time series, probability density function (*pdf*) of the calibrated model parameters, and a set of products to help evaluating convergence of the calibration process as well as the fit of the data.

[1] https://doi.org/10.1016/j.envsoft.2023.105761

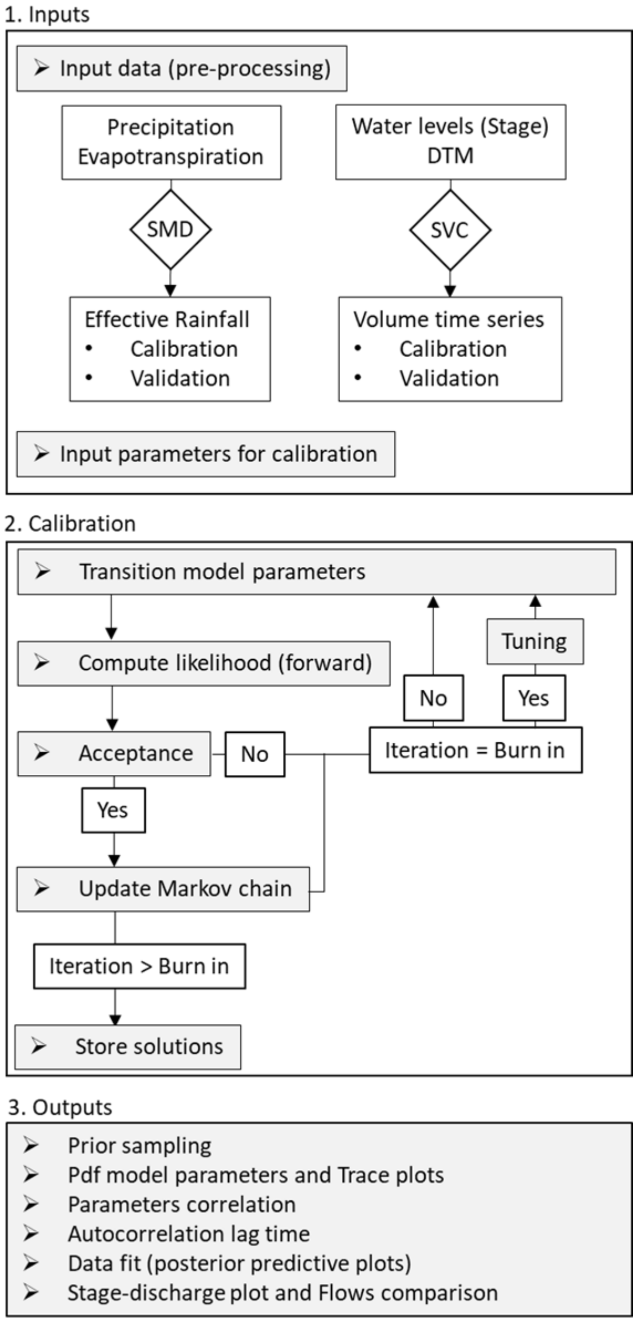

Figure 2 UISCEmod flowchart (from Campanyà et al. 2023). Gray boxes represent sub-modules, white boxes are related to conditions and steps within the sub-modules. SMD: soil moisture deficit model, SVC: stage volume conversion, DTM: digital terrain model.

For a more detailed description of UISCEmod please see Campanyà et al., (2023)[2].

[2] https://doi.org/10.1016/j.envsoft.2023.105761

2. INSTALLATION & REQUIREMENTS

UISCEmod can be downloaded from (https://github.com/joancampanya/UISCEmod). The general directory UISCEmod is divided into input (*in)*, output (*out*) and scripts (*scr*) directories. The input directory contains input data necessary to run UISCEmod, the output directory contains the products from UISCEmod, and the scripts directory contains the python scripts used to run UISCEmod.

The general directory UISCEmod can be copied to your computing facilities and, when the software requirements are satisfied, the code should find the paths automatically. Note that limitations on the maximum length of the paths may create issues, thus it is recommended adding UISCEmod general directory close to the local disk path.

In terms software requirements, although it may work with other versions, UISCEmod was developed using Spyder (Version: 5.1.5, Build: py39haa95532_1) in Anaconda (Version 2021.11, Build: py39_0) using Python 3 (Version: 3.9.7, Build: h6244533_1) in Windows 10.

The main libraries required to run UISCEmod include:

- Pandas (Version 1.3.4, Build: py39h6214cd6_0)
- Numpy (Version: 1.20.3, Build: py39ha4e8547_0)
- Matplotlib(Version: 3.4.3, Build: py39haa95532_0)
- Random(mkl_random, Version: 1.2.2, Build: py39hf11a4ad_0)
- Corner(Version: 2.2.1, Build: pypi_0, Channel: pypi)
- Scipy(Version: 1.7.1, Build: py39hbe87c03_2)

These libraries need to be installed before executing UISCEmod.

In addition to software requirements, the input datasets (Section 3) are also required.

3. INPUT FILES

All input files are within the "*in*" directory that is located within the UISCEmod general directory. Four inputs files are required to run UISCEmod, which are related to the following input information:

1. input time series
2. model parameters
3. job parameters
4. stage-area-volume curves

Comments can be added to any of the input files using the # symbol. Any text behind # will be ignored.

3.1 INPUT TIME SERIES

The input time series are included in a file named *"site name" + UISCEmod_input_ts.csv (e.g. site 1_UISCEmod_inputs_ts.csv)*, where site name is the reference name for the site. This file needs to be stores in the subdirectory "/UISCEmod/in/sites_input_ts/".

The information within this file includes: date, groundwater level in meters (*gwl[m]*), precipitation in millimeters (*R[mm]*), evapotranspiration in millimeters (*ET[mm]*), and evaporation in millimeters (*EVAP[mm]*).

The format of the file is:

```
Date,gwl[m],R[mm],ET[mm],EVAP[mm]
2023-01-01,25.5,0.0,0.47,0.68
2023-01-02,25.3,0.0,0.46,0.66
2023-01-03,,0.1,0.43,0.63
2023-01-04,25.8,0.3,0.49,0.68
…
```

where the first column is the date of the measurement in format "YYYY-MM-DD", *gwl* are the groundwater level time series, *R* is rainfall/precipitation time series, *ET* is evapotranspiration time series and *EVAP* is evaporation time series. Data gaps are only allowed in *gwl* time series.

3.2 MODEL PARAMETERS

The file with information about the model parameters has the following name structure “site name”_UISCEmod_info.csv and is located at “/UISCEmod/in/sites_info/” (e.g. /UISCEmod/in/sites_info/site_1_UISCEmod_info.csv).

This file contains information about the model parameters for specific site. The information is provided following the format: name of parameter followed by “:”. Then the potential values are specified within [] or by a single value. The main information provided by this file includes:

1) Data constrains to select the segments of the time series that need to be considered for calibration, validation and forward calculations.
2) Information relevant to generate potential effective rainfall time series.
3) Information about model parameters for the EM approach including range of possible values. Within this section, the information is provided as model_parameter:[*A*,*B*,*C*,*D*]. *A* is the minimum value valid for this model parameter, *B* is the maximum value valid for this model parameter, *C* is the starting value for this model parameter (* can be used to select a random value between *A* and *B*, assuming uniform distribution), *D* is True or False and specifies if this variable needs to be calibrated (True) or not (False). If not, the initial value *C* will be considered.

For more description about this file and the meaning and units of each parameter see the commented file within the examples provided.

3.3 JOB PARAMETERS

This file contains the jobs that need to be executed by UISCEmod. The name of this file is “job_parameters.csv” and it is in “/UISCEmod/in/sites_info/”. For each job the following information needs to be provided:

1) site_name: name of the site.
2) gen_ER: specify if UISCEmod needs to calculate effective rainfall time series (True) or not (False). If False, the potential ER time series need to be provided by external sources and be located in *“/UISCEmod/in/sites_input_ts/”* under the name *“site name” _ER_ts.csv (e.g. /UISCEmod/in/sites_input_ts/site_1_ER_ts.csv*). If True the file will be generated automatically based on the information from the model parameters file.
3) mode: specify the model approach to be considered (EM or LM)

4) version: extension to be added to the output directories.
5) iterations: number of iterations for the calibration process.
6) burn_in: proportion of iterations to consider before starting to save the accepted model parameters for posterior inference.
7) tunning: number of iterations to consider during tunning. Tunning is performed at burn_in. If the tunning value is negative it will not be performed.

Below there is an example of *job_parameters.csv*:

site_name,gen_ER,mode,model_approach,version,iterations,burn_in,tunning
Site 1,True,calibration,EM,_EM_v01,1e6,0.5,5000
Site 1,False,calibration,LM,_LM_v01,1e6,0.5,-3000

3.4 STAGE AREA VOLUME CURVE

This file contains the relationship between stage,area, and volume of the site. It needs to be located in "/UISCEmod/in/sites_info/" under the name "site name"_UISCEmod_SAV.csv (e.g. /UISCEmod/in/sites_info/Site 1_UISCEmod_SAV.csv). Below there is an example of the stage area volume curve file, where the first column is an index.

,Stage,Area,Volume
0,11.7,0.0,0.0
1,11.8,500.0,12.27836609
2,11.9,900.0,81.83708191
3,12.0,1300.0,206.0577393
4,12.1,1500.0,348.3259277
5,12.2,1600.0,506.3177185
…

4. OUTPUT FILES

UISCEmod output files are stored separately for each site and version in the "out" directory under the name site_name + version specified in the job parameters file (e.g. /UISCEmod/out/*Site_1_EM_v01/*). Within this directory the outputs are divided into calibration, validation and forward directories. Within each directory the output files are divided into *.csv* files, storing the calculated values, and *.png* files, to facilitate visualization of the datasets. List of main output files are described below:

4.1 CSV FILES

- **accepted_lik_cal_process.csv**: Likelihood values of the accepted model parameters.
- **accepted_model_parameters.csv**: Accepted model parameters after burn in.
- **accepted_ts_calibration_process.csv**: Volume time series for the accepted model parameters. If too many time series were accepted these will be resampled (max 500 time series) to avoid memory issues.
- **error_ts_calibration_process.csv**: errors of the accepted time series from the calibration process.
- **errors_quantiles.csv**: Amplitude of the empirically defined errors based on quantiles. This file is used to add the errors to the modelled time series during the forward calculations. The first column is the quantile and the second the variations from the median in [m^3].
- **model_volume_ts.csv**: Modelled volume time series using *pdf* of accepted model parameters.
- **model_stage_ts.csv**: Modelled stage time series using *pdf* of accepted model parameters. The first column is associated to the measured stage time series.
- **model_inflow_ts.csv**: Modelled inflow time series from the LM approach in m^3 using *pdf* of accepted model parameters. Only for the LM approach.
- **model_outflow_ts.csv**: Modelled outflow time series from LM approach in m^3 using *pdf* of accepted model parameters. Only for the LM approach.
- **model_overtop_ts.csv**: Modelled overtop time series from LM approach in m^3 using *pdf* of accepted model parameters. Only for the LM approach.
- **copy_input_file.csv**: A copy of the model parameters file from the inputs is stored in the out folder for reference.
- **tunned_step_size**: Comparison between original step size and step size after tunning for each model parameter.

4.2 PNG FILES

- **corner_plot_accepted_all.png**: corner plot using all accepted model parameters.
- **corner_plot_accepted_after_burn_in.png**: corner plot using model parameters accepted after burn in.
- **error_distribution_analysis.png**: Compare errors distribution between modeled and measured volume time series (in purple) with the calibrated sigma values (in black).
- **gamma_distribution.png**: Plots the gamma distribution of the accepted gamma parameters.
- **model_parameters_autocorrelation_analysis.png**: Plots autocorrelation analysis for model parameters. Can help estimate if the steps considered during the inversion process were too large or too small.
- **modelled_TS_empirical_errors_NSE_"X"_KGE_"X"_BIAS_"X"_"Y"_"Z"_volume.png**: Comparison between modelled and measured volume time series with summary of the results including NSE, KGE and BIAS. Y: refers to proportion of points within the 68% confidence interval and Z the proportion of points within the 95% confidence interval. The horizontal dashed lines within the plot represent the quantiles 5, 50 and 95 for modelled and measured time series.
- **modelled_TS_empirical_errors_NSE_"X"_KGE_"X"_BIAS_"X"_"Y"_"Z"_stage.png**: Comparison between modelled and measured stage time series with summary of the results including NSE, KGE and BIAS. Y: refers to proportion of points within the 68% confidence interval and Z the proportion of points within the 95% confidence interval. The horizontal dashed lines within the plot represent the quantiles 5, 50 and 95 for modelled and measured time series.
- **modelled_TS_model_param_propagation_NSE_"X"_KGE_"X"_BIAS_"X"_stage.png**: Comparison between model and measured stage time series defining the errors based on the distribution of the *pdf* of the model parameters.
- **modelled_TS_model_param_propagation_NSE_"X"_KGE_"X"_BIAS_"X"_volume.png**: Comparison between model and measured volume time series defining the errors based on the distribution of the *pdf* of the model parameters.
- **net_in_out_flows.png**: Include net flow from model time series and measured time series, modelled inflow, and modelled outflow.
- **stage_vs_net_flow_plot.png**: Plot modelled (blue) and measured (red) net flow versus stage. For the LM approach the discharge (black) vs stage is also included.
- **trace_and_distribution_plots.png**: Show trace plots and distribution of all model parameters considered for calibration. When sampling the prior the extension "_prior_sampling" will be added. If only one value is accepted for a certain parameter this will not be added to the plot.

5. UISCEMOD SCRIPT FILES

UISCEmod is divided in five python files described below:

- *UISCEmod_v072023.py*: This is the main code that needs to be executed in order to run UISCEmod.
- *UISCEmod_library_v072023.py*: Contains classes and functions used to run UISCEmod. Functions are divided into the following classes: inputs, preprocessing, forwards, mcmc, outputs, and visualization.
- *UISCEmod_module_inputs_v072023.py*: This module preprocess input time series for UISCEmod and generate input variables for calibration and output modules.
- *UISCEmod_module_calibration_v072023.py*: Calibrates the hydrological models. The calibration process is automated following a Bayesian approach.
- *UISCEmod_module_outputs_v072023.py*: Generates the main products from UISCEmod including: model stage and volume time series, probability density function (*pdf*) of the calibrated model parameters, and a set of products to help evaluating convergence of the calibration process as well as the fit of the data.

The user does not need to modify any of these files as the information to run the jobs is extracted from the input files.

5.1 MODULES

UISCEmod code is divided into the three modules presented below. Further details about the modules and functions are available within the associated python files.

5.1.1 INPUTS

Gets inputs time series and pre-processes the data to be ready for calibration and outputs. The main functions executed in this module are:

- Get details about the job to perform.
  - Function: *uisce.inputs.get_job_details*
- Get initial model parameters and range of variability for the model parameters from "site name" + _UISCEmod_info.csv.
  - Function: *uisce.inputs.get_model_parameters*
- Get meteorological data and calculate/read effective rainfall time series to consider.
  - Function: *uisce.inputs.get_meteorological_datasets*
- Get measured groundwater level time series to use for calibration and validation.

    - Function: *uisce.inputs.get_water_level_data*
- Convert groundwater level time series to volume time series and divide the data into calibration, validation and forward datasets.
    - Function: *uisce.preprocessing.initial_conditions*
- Generate datasets to be used for calibration/forward process.
    - Function: *uisce.preprocessing.model_parameters*

5.1.2 CALIBRATION

Calibration of the model parameters is automated following a Bayesian approach to constrain the model parameters that are more likely to explain the data. The main functions considered within the calibration module are:

- Define initial variables required for calibration process
    - Function: *uisce.mcmc.define_initial_variables*

Start the calibration process:

- Tunning. This will only activate when appropriate, after burn-in. It is used to modify the step size in order to improve the sampling of the *pdf*. The step sizes to be assessed during tuning are specified in the *get_paths_and_initial_information* function. The user can modify the potential step sized based on the case of study.
    - Function: *uisce.mcmc.tunning*
- Transition between model parameters. Generate new model parameters (candidates). At this stage the new model parameters may or may not be accepted in the McMC chain
    - Function: *uisce.mcmc.transition_model_parameters*
- Check for convergence. Check if the calibration process is converging. If the calibration is still not converging UISCEmod may restart the calibration process with random values instead of the suggested initial values. The user may change the thresholds if considered appropriate.
    - Function: *uisce.mcmc.check_convergence*
- Check that the model parameters agree with the prior.
    - Function: *uisce.mcmc.prior*
- Export likehood of the suggested model parameters and the associated time series
    - Function: *uisce.mcmc.lik_normal*
- Acceptance/Rejection of the suggested model parameters. If not accepted they will be stored as rejected values.
    - Function: *uisce.mcmc.acceptance*

- If the suggested model parameters are accepted the McMC chain is updated and the model parameters and likelihood values are stored.
    - Function: *uisce.evaluation.store_accepted_parameters*
    - Function: *uisce.mcmc.keep_time_series*
- Visualize evolution of the calibration process with corner plots.
    - Function: *uisce.visualization.corner_plot*
- At the end of the calibration process check proportion of accepted and rejected parameters before and after the burn in.
    - Function: *uisce.mcmc.proportion_accepted_models*

5.1.3 OUTPUTS

The outputs module performs the forward calculations and assesses the confidence in the results by providing relevant information related to: 1) adequate sampling of the prior $p(\theta)$, 2) convergence of the calibration process and correlation between model parameters, 3) distribution of the error residuals, and 4) data fit.

- Visualize dependency between variables with corner plots.
    - Function: *uisce.visualization.variables_dependency*
- Reformat accepted model parameters to facilitate visualization.
    - Function: *uisce.outputs.reformat_accepted_parameters*
- Update and store accepted time series.
    - Function: *uisce.outputs.store_accepted_ts*
- Store accepted model parameters.
    - Function: *uisce.outputs.store_accepted_parameters*
- Assess error distribution and compare with the constrained sigma parameter.
    - Function: *uisce.outputs.errors_analysis*
- Visualize the results and generate relevant products including: 1) distribution of the model parameters and trace plots, 2) accepted gamma curves, 3) modelled volume and stage time series considering only the *pdf* of the accepted model parameters and including the empirically estimated errors, 4) stage-discharge curves, 5) inflow, net-flow and outflow times series (only for LM approach), 6) autocorrelation analysis for accepted model parameters.
    - Function: *uisce.outputs.results_and_products*

6. EXECUTING UISCEMOD

In order to execute UISCEmod the user does not need to modify any of the python scripts from *scr* directory, except if implementing updates or adapting the code to new scenarios.

To run UISCEmod the following points need to be addressed:

1) Download UISCEmod general directory (https://github.com/joancampanya/UISCEmod) and place it to your computer facilities.
2) Install the software requirements (Section 2)
3) Generate input files (Section 3). Examples of input files are provided.
4) Execute *UISCEmod_v072023.py*. Remember to uncomment the "random.seed(72023)" at the beginning of *UISCEmod_library_v072023.py* if you want to compare the results with the examples provided.

If steps 1 and 2 are performed, UISCEmod can be executed with the example datasets (Section 7) and the results compared with the provided in the example to assess if UISCEmod is working as expected. If running multiple jobs simultaneously, with an updated *job_parameters.csv* file, make sure to add different extension to the jobs by modifying the job_ext variable in *UISCEmod_v072023.py* file.

7. UISCEMOD EXAMPLE

An example is provided for a specific site for the EM and LM approaches. You can use this example to test that UISCEmod is working as expected. The provided examples were provided for a limited number of iterations to reduce the computational times, which means that the calibration process was not completed for neither of them. In order to be able to compare your results with the results provided in the examples you need to uncomment the command "random.seed(72023)" at the beginning of *UISCEmod_library_v072023.py* file (~line 32). Do not forget to comment this line when working with your sites. Also, when comparing with the example provided do not modify any of the input files.

When UISCEmod is working as expected, feel free to modify the input files (e.g. increase the number of iterations within *job_parameters.csv*, apply tunning to the LM approach) to allow UISCEmod to improve the calibration of the model parameters for the example datasets. The user can check if the results (NSE, KGE and BIAS) of the new calibration processes are similar to those provided by Campanyà et al. (2023)[3] for validation and calibration datasets for the site included in the example.

[3] https://doi.org/10.1016/j.envsoft.2023.105761